Технології програмування

Лабораторна робота №3

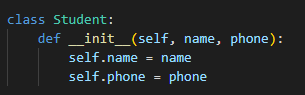
**Робота з файлами. Юніт тести.**

**Мета роботи:** Використовуючи теоретичне підґрунтя про ООП у мові Python переробити програму телефонного довідника студентів використовуючи принципи ООП для формування відомостей про студентів.

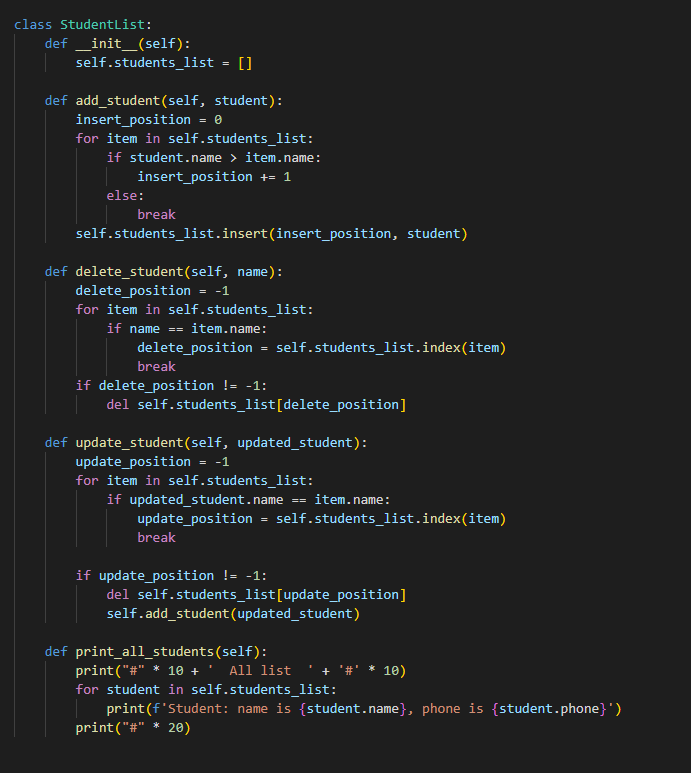
**Хід роботи**

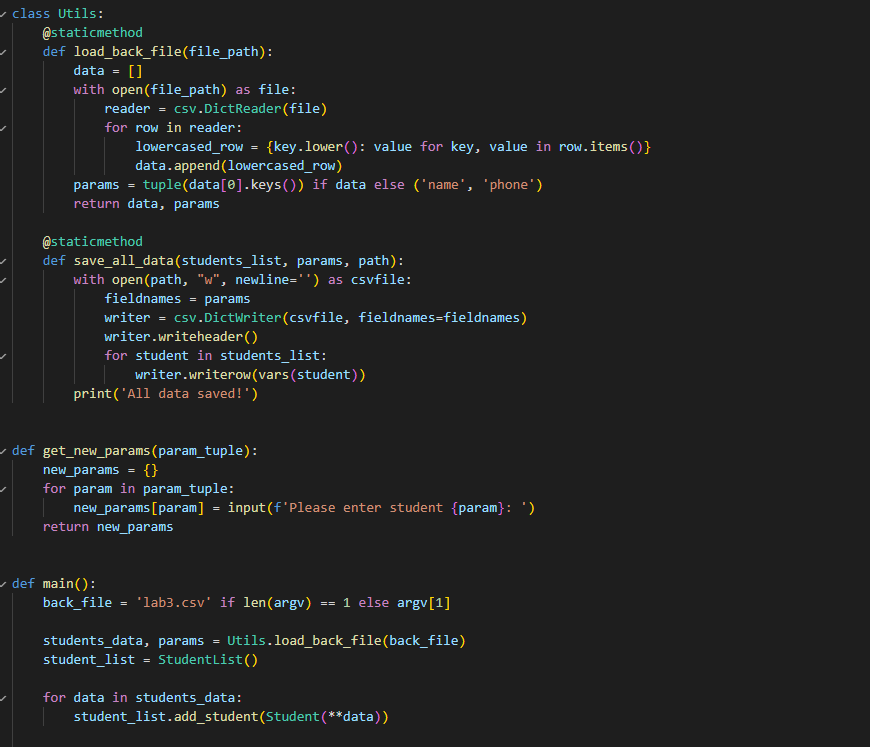
**Переробити** функціональність телефонного довідника студентів групи, що був розроблений у Лабораторній роботі №2 використовуючи принципи ООП:

1)розробити клас Студент групи з відповідними атрибутами;



2)розробити клас Список групи, має містити не словники, як виконано в лабораторній роботі №2, а об’єкти класу Студент групи; додавання нового запису, видаленні існуючого чи зміна даних має бути виконана через методи класу Список групи.

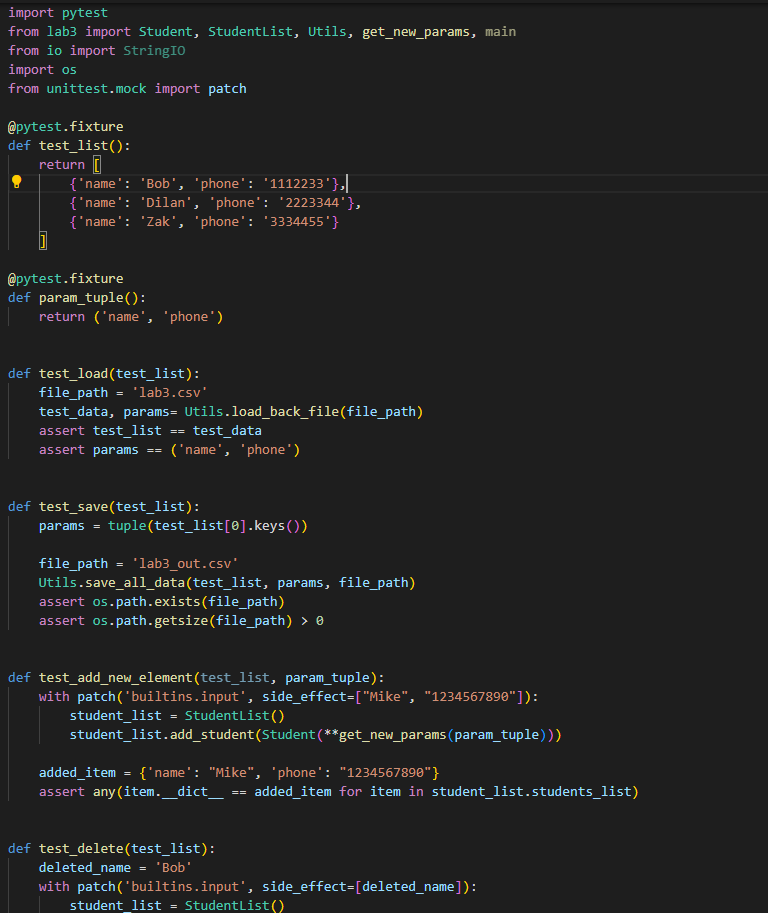


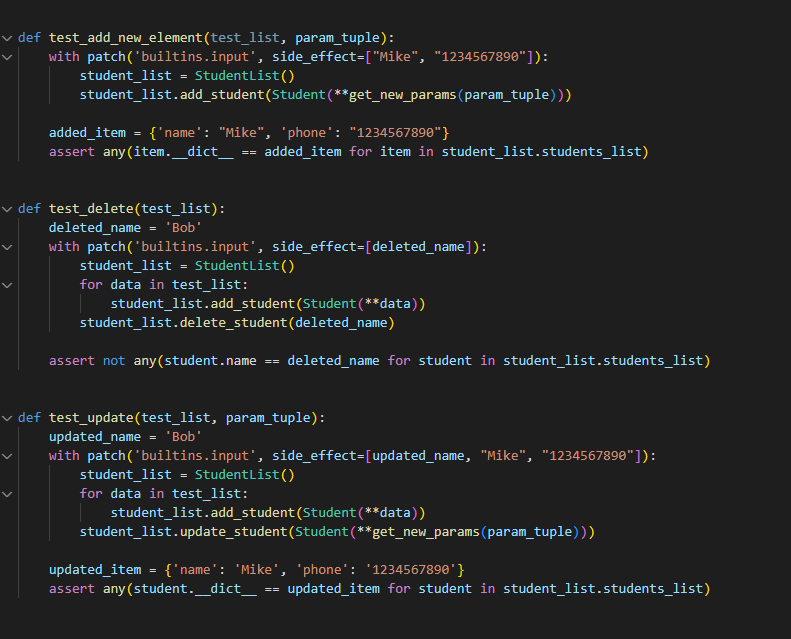
3)розробити клас для роботи з файлами для зчитування початкової інформації про список групи та збереження інформації по завершенню програми. 

4)список студентів має містити не словники, як виконано в лабораторній роботі №2, а об’єкти класу Студент групи;

5)описання всіх класів мають міститися в окремих файлах, що мають відповідні імена(наприклад Studen, StudentList, Utils)

6)основний функціонал програми має бути покритий Юніт тестами.





**Текст програми разом зі звітом розмістити в директорії lab\_03.**

Висновок: В ході переробки програми телефонного довідника студентів було успішно використано принципи об'єктно-орієнтованого програмування (ООП) у мові Python. ООП спрямований на організацію коду у вигляді об'єктів, які об'єднують дані та методи, що опрацьовують ці дані.

У першу чергу було створено клас `Student`, який представляє студента з атрибутами `name` та `phone`. Цей клас дозволяє більш ефективно та логічно моделювати дані про студентів, а використання екземплярів цього класу робить код більш зрозумілим та підтримуваним.

Далі був створений клас `StudentList`, який представляє список студентів та має методи для додавання, видалення та оновлення студентів. Використання цього класу дозволяє групувати функціонал, пов'язаний із списком студентів, що полегшує обслуговування та збереження коду.

Також був створений клас `FileHandler`, який відповідає за роботу з файлами – зчитування початкової інформації та збереження оновленої інформації. Це дозволяє відокремити логіку роботи з файлами від основної логіки програми, що робить код більш гнучким та повторно використовуваним.

Кожен клас має відповідальність за конкретний аспект програми, а взаємодія між ними відбувається через обмін об'єктами. Це дозволяє досягти високого рівня абстракції та підтримуваності програми. В результаті використання принципів ООП програма стала більш структурованою, зрозумілою та готовою до подальших розширень.